

Sayson 390/480 ADSI phones

**Asterisk Developer
Information/FAQs**

Contents

- Additional Documentation**4
 - Digium’s web site4
 - The ADSI Belcore/Telcordia Specs4
- General Information**5
 - Asterisk Built in ADSI Scripting Tool5
 - Web Configurator Service5
 - Asterisk Forum’s5
 - FDN and Security Codes6
 - Explanation of the Slots in Services6
 - ADSI type 1 Syntax7
 - Event List8
- Commedian Mail**9
- Samples**10
- FAQ**11

Sayson Technologies Ltd. will not accept liability for any damages, or long distance charges, which may result from unauthorized and/or unlawful use.

While every effort has been made to ensure accuracy, Sayson Technologies Ltd. will not be liable for technical or editorial errors or omissions contained within this document. The information contained in this document is subject to change without notice.

Sayson Technologies, Ltd.
www.sayson.com

Copyright 2004 Sayson Technologies Ltd.
All Rights Reserved.

Introduction

Information provided here is intended as a guideline for Asterisk developers who wish to develop Sayson 390 or 480 ADSI analog phones for the Asterisk platform. The information provided here was gathered from various open sources or has been submitted or researched from submissions from other Asterisk developers or user groups or through Sayson's own programming experience.

Additional Documentation

Digium's web site:

<http://www.digium.com/index.php?menu=adsi>

The ADSI Belcore/Telcordia Specs

Available from Telcordia are:

- Document: SR-2461 - (A Module of ADSI, FR-12)
Title: Customer Premises Equipment Compatibility Guidelines for the Analog Display Services Interface (ADSI)
- Document: GR-1273-CORE - (A Module of LSSGR, FR-64 and ADSI, FR-12)
Title: Analog Display Services Interface (ADSI) SPCS/Server Generic Requirements

Note: The ADSI Belcore/Telcordia specs are not necessary to develop scripts

General Information

Asterisk Built in ADSI Scripting Tool

The Asterisk system has its own built in ADSI scripting tool that allows developers to create their own phone programming and download it into ADSI phones such as the Sayson 390/480 phones through the system itself. A small section of text needs to be included in the preamble of an ADSI file.

For more information on these ADSI phones go to digium's website:

<http://www.digium.com/index.php?menu=adsi>

Web Configurator Service

If you don't want to develop your own ADSI script through the Asterisk's built in ADSI scripting tool, Sayson 390/480 phones have been set up so that they can also be programmed from a free Web Service at www.sayson.com/dealer. This service has isn't as diverse a tool as the Asterisk's tool kit, but is easier to use – no programmer's coding required. To access this page, contact Sayson for your user name and password.

- Simple Mode - None state sensitive: http://www.sayson.com/dealer/simple_instructions.htm
- Advanced Mode - State sensitive (Menus that detect triggers for Idle, Connected, and Call Waiting (through Call Waiting Caller ID information): http://www.sayson.com/dealer/advanced_mode.htm

Note: The Sayson Web Configurator is easy to use, but it does limit you to 3 main menus of customizable softkeys without the option to build submenus within those softkeys.

Asterisk Forums

A great resource for questions relating to the Asterisk and ADSI is the Asterisk mailing lists. For more information, go to <http://lists.digium.com/mailman/listinfo/>.

Programmer's Information

FDN and Security Codes

If you are developing a script through the Asterisk's built in ADSI tool, then the feature download number (FDN) and security access codes must be included at the beginning of a script. The FDN determines which memory slot the script will be downloaded into, and the security number allows access to this slot. Without this, the phone won't know which slot to download the programming into and won't accept it if the security number is wrong. This information needs to be included in the preamble of the script.

Slot 1

```
FDN: 0.0.0.F
SECURITY: 9B.DB.F7.AC
```

Slot 2

```
FDN: 85.EF.D9.DA
SECURITY: 78.92.1D.49
```

Slot 3

```
FDN: 7B.C6.45.0C
SECURITY: 9B.60.94.30
```

Slot 4

```
FDN: FE.2E.A5.D1
SECURITY: 79.A9.0C.F0
```

Explanation of the Slots in Services

Slot 1

This is the only slot that should contain programming in order for a script to be active during regular phone activities. The programming in this slot is triggered when the phone detects call activity, for example, when the phone rings or is placed off-hook.

Slot 2

This is the Self Launching slot for these phones and is triggered when the phone has had no call activity for a certain amount of time. On all 390 and older 480 phones this, amount of time is 6-7 seconds. On new 480 phones, this amount of time is almost immediate. The programming downloaded into this slot can be identical to slot 1, or it can be completely different, such as for advertising purposes.

Slots 3 and 4

The only way these can be activated is by pressing the "Services" button on your phone and scrolling down and selecting a slot. Slots 3 and 4 cannot be activated this way while the phone is active, for example, when on a call.

Note: Whenever the "goodbye" button is pressed or if the phone placed on-hook, the phone will exit the programming and go back to it's default screen until the programming is triggered again.

For more information, go to <http://www.digium.com/index.php?menu=adsi>.

ADSI type 1 Syntax

Found in the source file for ADSI type 1:

```
static struct adsi_key_cmd kcmds[] = {
    { "SENDDTMF", 0, send_dtmf },
    /* Encoded DTMF would go here */
    { "ONHOOK", 0x81 },
    { "OFFHOOK", 0x82 },
    { "FLASH", 0x83 },
    { "WAITDIALTONE", 0x84 },
    /* Send line number */
    { "BLANK", 0x86 },
    { "SENDCHARS", 0x87 },
    { "CLEARCHARS", 0x88 },
    { "BACKSPACE", 0x89 },
    /* Tab column */
    { "GOTOLINE", 0x8b, goto_line },
    { "GOTOLINEREL", 0x8c, goto_line_rel },
    { "PAGEUP", 0x8d },
    { "PAGEDOWN", 0x8e },
    /* Extended DTMF */
    { "DELAY", 0x90, send_delay },
    { "DIALPULSEONE", 0x91 },
    { "DATAMODE", 0x92 },
    { "VOICEMODE", 0x93 },
    /* Display call buffer 'n' */
    /* Clear call buffer 'n' */
    { "DIGITCOLLECT", 0x96, digitcollect },
    { "DIGITDIRECT", 0x96, digitdirect },
    { "CLEAR", 0x97 },
    { "SHOWDISPLAY", 0x98, showdisplay },
    { "CLEARDISPLAY", 0x98, cleardisplay },
    { "SHOWKEYS", 0x99, showkeys },
    { "SETSTATE", 0x9a, set_state },
    { "TIMERSTART", 0x9b, starttimer },
    { "TIMERCLEAR", 0x9b, cleartimer },
    { "SETFLAG", 0x9c, setflag },
    { "CLEARFLAG", 0x9c, clearflag },
    { "GOTO", 0x9d, subscript },
    { "EVENT22", 0x9e },
    { "EVENT23", 0x9f },
    { "EXIT", 0xa0 },
}
```

```

static struct adsi_key_cmd opcmds[] = {
    /* 1 - Branch on event - handled specially */
    { "SHOWKEYS", 2, showkeys },
    /* Display Control */
    { "SHOWDISPLAY", 3, showdisplay },
    { "CLEARDISPLAY", 3, cleardisplay },
    { "CLEAR", 5 },
    { "SETSTATE", 6, set_state },
    { "TIMERSTART", 7, starttimer },
    { "TIMERCLEAR", 7, cleartimer },
    { "ONEEVENT", 8, oneevent },
    /* 9 - Subroutine label, treated specially */
    { "SETFLAG", 10, setflag },
    { "CLEARFLAG", 10, clearflag },
    { "DELAY", 11, send_delay },
    { "EXIT", 12 },
};

```

Event List

1. CallerID
2. VMWI - (Visual Message Waiting Indicator)
3. Answer
4. FarEndAnswer
5. EndofRinging
6. SelectService
7. Pickup
8. CIDCW - (Caller ID in Call Waiting)
9. FarEndBusy
10. FarEndRinging
11. Dialtone
12. RecallDialTone
13. MessageWaitingTone
14. Reorder
15. DistinctiveRinging
16. NormalRinging
17. RingReminder
18. SpecialRinging
19. CodedRinging
20. Timer - (expires)
21. ExtensionInUse
22. CustomEvent1 - (return string control character x(9E))
23. CustomEvent2 - (return string control character x(9F))
24. CPEID (received)

Comedian Mail

If you are using ADSI phones and trying to access Comedian Mail, Comedian Mail tries to do an FDM download (download a new script) to the phone first. If you don't change the FDN and security codes in the Comedian Mail app, when trying to access voicemail you will get the error message:

```
Comedian Mail
download refused

Services is full
```

In the “app_voicemail.c” file, the “adsi_begin_download” is invoked as follows:

```
if (adsi_begin_download(chan, addesc, adapp, adsec, adver)) {
```

where “adapp” (fdn) and “adsec” are hardcoded as follows:

```
static char *adapp = "CoMa";
static char *adsec = "_AST";
```

These need to be changed to the correct FDN and Security numbers for the slot you wish to download this programming into. Since you have probably already created your own FDN script for the phones, you won't want the Comedian Mail to overwrite this. A suggestion would be to use the FDN and Security numbers for a slot you are not currently using, such as slot 3 or 4. As an example, if you were to enter this information for slot 2, you would enter the FDN and Security information as:

```
static char *adapp = "\x85\xEF\xD9\xDA";
static char *adsec = "\x78\x92\x1D\x49";
```

Then, recompile and restart app_voicemail.

Samples

The Asterisk ADSI tool kit should come with a sample ADSI type 1 script, shown below:

```
;
; Asterisk default ADSI script
;
;
; Begin with the preamble requirements
;
DESCRIPTION "Asterisk PBX"           ; Name of vendor
VERSION 0x02                         ; Version of stuff
;SECURITY "_AST"                     ; Security code
SECURITY 0x0000                      ; Security code
FDN 0x0000000f                      ; Descriptor number

;
; Flags
;
FLAG "nocallwaiting"

;
; Predefined strings
;
DISPLAY "titles" IS "** Asterisk PBX **"
DISPLAY "talkingto" IS "Call active." JUSTIFY LEFT
DISPLAY "callname" IS "$Call1p" JUSTIFY LEFT
DISPLAY "callnum" IS "$Call1s" JUSTIFY LEFT
DISPLAY "incoming" IS "Incoming call!" JUSTIFY LEFT
DISPLAY "ringing" IS "Calling... " JUSTIFY LEFT
DISPLAY "callended" IS "Call ended." JUSTIFY LEFT
DISPLAY "missedcall" IS "Missed call." JUSTIFY LEFT
DISPLAY "busy" IS "Busy." JUSTIFY LEFT
DISPLAY "reorder" IS "Reorder." JUSTIFY LEFT
DISPLAY "cwdisabled" IS "Callwait disabled"
DISPLAY "empty" IS "asdf"

;
; Begin soft key definitions
;
KEY "callfwd" IS "CallFwd" OR "Call Forward"
    OFFHOOK
    VOICEMODE
    WAITDIALTONE
    SENDDTMF "*60"
    GOTO "offHook"
ENDKEY
```

```
KEY "vmail_OH" IS "VMail" OR "Voicemail"
    OFFHOOK
    VOICEMODE
    WAITDIALTONE
    SENDDTMF "8500"
ENDKEY
```

```
KEY "vmail" IS "VMail" OR "Voicemail"
    SENDDTMF "8500"
ENDKEY
```

```
KEY "backspace" IS "BackSpc" OR "Backspace"
    BACKSPACE
ENDKEY
```

```
KEY "cwdisable" IS "CWDsble" OR "Disable Call Wait"
    SENDDTMF "*70"
    SETFLAG "nocallwaiting"
    SHOWDISPLAY "cwdisabled" AT 4
    TIMERCLEAR
    TIMERSTART 1
ENDKEY
```

```
KEY "cidblock" IS "CIDBlk" OR "Block Callerid"
    SENDDTMF "*67"
    SETFLAG "nocallwaiting"
ENDKEY
```

```
;  
; Begin main subroutine  
;
```

```
SUB "main" IS
    IFEVENT NEARANSWER THEN
        CLEAR
        SHOWDISPLAY "titles" AT 1 NOUPDATE
        SHOWDISPLAY "talkingto" AT 2 NOUPDATE
        SHOWDISPLAY "callname" AT 3
        SHOWDISPLAY "callnum" AT 4
        GOTO "stableCall"
    ENDIF
    IFEVENT OFFHOOK THEN
        CLEAR
        CLEARFLAG "nocallwaiting"
        CLEARDISPLAY
        SHOWDISPLAY "titles" AT 1
        SHOWKEYS "vmail"
        SHOWKEYS "cidblock"
```

```

        SHOWKEYS "cwdisable" UNLESS "nocallwaiting"
        GOTO "offHook"
ENDIF
IFEVENT IDLE THEN
    CLEAR
    SHOWDISPLAY "titles" AT 1
    SHOWKEYS "vmail_OH"
ENDIF
IFEVENT CALLERID THEN
    CLEAR
;    SHOWDISPLAY "titles" AT 1 NOUPDATE
;    SHOWDISPLAY "incoming" AT 2 NOUPDATE
    SHOWDISPLAY "callname" AT 3 NOUPDATE
    SHOWDISPLAY "callnum" AT 4
ENDIF
IFEVENT RING THEN
    CLEAR
    SHOWDISPLAY "titles" AT 1 NOUPDATE
    SHOWDISPLAY "incoming" AT 2
ENDIF
IFEVENT ENDOFRING THEN
    SHOWDISPLAY "missedcall" AT 2
    CLEAR
    SHOWDISPLAY "titles" AT 1
    SHOWKEYS "vmail_OH"
ENDIF
IFEVENT TIMER THEN
    CLEAR
    SHOWDISPLAY "empty" AT 4
ENDIF
ENDSUB

SUB "offHook" IS
    IFEVENT FARRING THEN
        CLEAR
        SHOWDISPLAY "titles" AT 1 NOUPDATE
        SHOWDISPLAY "ringing" AT 2 NOUPDATE
        SHOWDISPLAY "callname" at 3 NOUPDATE
        SHOWDISPLAY "callnum" at 4
    ENDIF
    IFEVENT FARANSWER THEN
        CLEAR
        SHOWDISPLAY "talkingto" AT 2
        GOTO "stableCall"
    ENDIF
    IFEVENT BUSY THEN
        CLEAR
        SHOWDISPLAY "titles" AT 1 NOUPDATE
        SHOWDISPLAY "busy" AT 2 NOUPDATE

```

```

        SHOWDISPLAY "callname" at 3 NOUPDATE
        SHOWDISPLAY "callnum" at 4
    ENDIF
    IFEVENT REORDER THEN
        CLEAR
        SHOWDISPLAY "titles" AT 1 NOUPDATE
        SHOWDISPLAY "reorder" AT 2 NOUPDATE
        SHOWDISPLAY "callname" at 3 NOUPDATE
        SHOWDISPLAY "callnum" at 4
    ENDIF
ENDSUB

SUB "stableCall" IS
    IFEVENT REORDER THEN
        SHOWDISPLAY "callended" AT 2
    ENDIF
ENDSUB

```

Sayson develops scripts using ADSI type 2. ADSI type 2 scripts cannot currently be written or downloaded from the Asterisk system. Although the syntax will be slightly different, a sample of the ADSI type 2 code is included here as an example of the structure and flow of a script:

```

;-----
;
;   Date       : 17:03:13 12/31/2003
;   Author      : CJR
;   Company     : Sayson Technologies Ltd.
;   Notes       : This file is ADSI Type 2 and cannot be used on the
;                  built-in tool kit on the Asterisk open source PBX,
;                  which adhere's to ADSI Type 1.
;
;               This file is intended as a developers aide on how to
;                  structure a script. - Sayson
;
;-----

;-----
;   Line Definitions
;-----

DefineLine Line01=Normal,Center,WrapOFF,
DefineLine Line02=Normal,Center,WrapOFF,          ; Title of PBX
DefineLine Line03=Normal,Center,WrapOFF,
DefineLine Line04=Normal,Center,WrapOFF,
DefineLine Line05=Normal,Center,WrapOFF,V 2.1.3.02.28a

```

```

DefineLine Line06=Normal,Center,WrapOFF,Programmed by
DefineLine Line07=Normal,Center,WrapOFF,Sayson
DefineLine Line08=Normal,Center,WrapOFF,www.sayson.com
DefineLine Line09=Normal,Center,WrapOFF,
DefineLine Line10=Normal,Center,WrapOFF,<PHN> DND enabled
DefineLine Line11=Normal,Center,WrapOFF,(2 of 2)
DefineLine Line12=Normal,Left,WrapOFF,Inbound Call:
DefineLine Line13=Normal,Left,WrapOFF,Connected to:
DefineLine Line14=Normal,Left,WrapOFF,Call Waiting...
DefineLine Line15=Normal,Left,WrapON,$Call1s<TAB>$Call1p ; Caller ID name
DefineLine Line16=Normal,Left,WrapON,$Call2s<TAB>$Call2p ; Caller ID number
DefineLine Line17=Normal,Left,WrapOFF,Manual Operation
DefineLine Line18=Normal,Center,WrapON,Please choose a Call<TAB>Transfer Option.
DefineLine Line19=Normal,Center,WrapON,Enter Ext or Tel<TAB>number and press OK.
DefineLine Line20=Normal,Center,WrapON,Enter Call Number(s)<TAB>and press OK.
DefineLine Line21=Normal,Center,WrapON,Enter Call Number<TAB>and press OK.
DefineLine Line22=Normal,Center,WrapON,Please hangup when <TAB>finished.
DefineLine Line23=Normal,Center,WrapON,Please choose a<TAB>Conferencing Option.
DefineLine Line24=Normal,Center,WrapOFF,(Use "#" inbetween)
DefineLine Line25=Normal,Center,WrapOFF,In Conference
DefineLine Line26=Normal,Center,WrapON,Enter Extension<TAB>number and press OK.
DefineLine Line27=Normal,Center,WrapON,Enter Password<TAB>and press OK.
DefineLine Line28=Normal,Center,WrapON,Dial Ext or Tel<TAB>No. followed by "#"
DefineLine Line29=Normal,Center,WrapOFF,Hangup when done.
DefineLine Line30=Normal,Left,WrapOFF,1 Manage Voice Msgs
DefineLine Line31=Normal,Left,WrapOFF,2 Manage Faxes
DefineLine Line32=Normal,Left,WrapOFF,3 Manage Email
DefineLine Line33=Normal,Left,WrapOFF,4 Send Voice Mail
DefineLine Line34=Normal,Left,WrapOFF,5 Personal Options
DefineLine Line35=Normal,Center,WrapON,Enter Ext. number, <TAB>passcode and then #.
DefineLine Line36=Normal,Center,WrapOFF,Follow the prompts.
DefineLine Line37=Normal,Center,WrapOFF,Calls on Hold
DefineLine Line39=Normal,Center,WrapON,Enter Ext or Tel<TAB>number, then "#".
DefineLine Line40=Normal,Center,WrapOFF,Hangup when done.
DefineLine Line41=Normal,Center,WrapON,Enter Call Number,<TAB>Ext or Tel number
DefineLine Line42=Normal,Center,WrapOFF,and then "#".
DefineLine Line43=Normal,Center,WrapON,Enter Call Number(s)<TAB>and press "#".
DefineLine Line45=Normal,Center,WrapOFF, Exits slot 2

```

```

;-----
;   SoftKey Definitions
;-----

```

```

;-----
;   Label:   Answer
;   Comment: (Ringing State)
;-----

```

```

SoftKey=SK_01
Label=Answer

```

```
        SetHookState OffHook
EndSoftKey
```

```
;-----
;   Label:   Flash
;   Comment: (Connected State) Flash Softkey
;-----
SoftKey=SK_02
    Label=Flash
    LongLabel=Flash/Tsfr/Conf
    SetHookState Link
EndSoftKey
```

```
;-----
;   Label:   Hangup
;   Comment: (Connected State)
;-----
SoftKey=SK_03
    Label=Hangup
    SetState 1
    SetInfoFlags 3
    SetHookState OnHook
    Delay 25
    JumpTo #00
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Idle State) Back to Idle Menu
;-----
SoftKey=SK_05
    Label=Back
    SetState 1
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
    JumpTo #00
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Idle State) Back to Idle Menu 2
;-----
SoftKey=SK_06
    Label=Back
    SetState 1
```

```
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
    JumpTo IdleState2
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Connected State) Back to Connected Menu (1 x Flash)
;-----
```

```
SoftKey=SK_07
    Label=Back
    SetState 4
    SetInfoFlags 3
    SetHookState Link
    ClearCB 2
    Keycode ClearChars
    JumpTo #00
    SetEvent 2
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Connected State) Back to Connected Menu (no Flash)
;-----
```

```
SoftKey=SK_08
    Label=Back
    SetState 4
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
    JumpTo #00
    SetEvent 2
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Connected State) Back to Connected Menu 2 (no Flash)
;-----
```

```
SoftKey=SK_09
    Label=Back
    SetState 4
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
```

```
        JumpTo ConnectedState2
        SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   BkSpc
;-----
SoftKey=SK_10
    Label=BkSpc
    Keycode Backspace
    Keycode=Voice
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Connected State) Transfer (back to transfer menu)
;-----
SoftKey=SK_11
    Label=Back
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
    JumpTo Transfer
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Back
;   Comment: (Connected State) Conference (back to conference menu)
;-----
SoftKey=SK_12
    Label=Back
    SetInfoFlags 3
    ClearCB 2
    Keycode ClearChars
    JumpTo Conf
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Transfr
;   Comment: (Connected State)
;-----
SoftKey=SK_13
    Label=Transfr
```

```
    SetState 3
    JumpTo Transfer
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Current
;   Comment: (Connected State) Transfer
;-----
```

```
SoftKey=SK_14
    Label=Current
    SetHookState Link
    Delay 100
    String 1
    ClearCB 2
    JumpTo TransferCurrentCall
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Select
;   Comment: (Connected State) Transfer
;-----
```

```
SoftKey=SK_15
    Label=Select
    SetHookState Link
    Delay 100
    String 2
    ClearCB 2
    JumpTo TransferSelectCall
    SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Conf.
;   Comment: (Connected State)
;-----
```

```
SoftKey=SK_16
    Label=Conf.
    SetState 3
    JumpTo Conf
    SetEvent 1
EndSoftKey
```

```
;-----
```

```
; Label: Select
; Comment: (Connected State) Conference
;-----
SoftKey=SK_17
Label=Select
SetHookState Link
Delay 100
String 4
ClearCB 2
JumpTo ConfSelect
SetEvent 1
EndSoftKey
```

```
;-----
; Label: All
; Comment: (Connected State) Conference
;-----
SoftKey=SK_18
Label=All
SetHookState Link
Delay 100
String 3
ClearCB 2
JumpTo HangupScreenCE1andInConfCE2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: NewCall
; Comment: (Connected State)
;-----
SoftKey=SK_19
Label=NewCall
SetInfoFlags 11
SetState 3
JumpTo NewCall
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Connected State) New Call OK
;-----
SoftKey=SK_20
Label=OK
SetState 1
```

```
SetInfoFlags 3
SetHookState Link
Delay 100
String 6
Delay 100
Keycode SendChars
Keycode ClearChars
String # ; Does this require a 1 or # even?
String 1
ClearCB 2
JumpTo #00
SetEvent 2
EndSoftKey
```

```
;-----
; Label: UnHold
; Comment: (Connected State)
;-----
SoftKey=SK_21
Label=UnHold
SetInfoFlags 3
SetHookState Link
Delay 100
String 7
ClearCB 2
JumpTo #00
SetEvent 2
EndSoftKey
```

```
;-----
; Label: Discnct
; Comment: (Connected State)
;-----
SoftKey=SK_22
Label=Discnct
SetInfoFlags 3
SetHookState Link
Delay 100
String 8
ClearCB 2
JumpTo #00
SetEvent 2
EndSoftKey
```

```
;-----
; Label: More
```

```
;    Comment: (Connected State) More to Connected State 2
;-----
SoftKey=SK_23
    Label=More
    SetState 3
    JumpTo ConnectedState2
    SetEvent 1
EndSoftKey
```

```
;-----
;    Label:    Pickup
;    Comment: (Idle State)
;-----
SoftKey=SK_24
    Label=Pickup
    SetInfoFlags 3
    SetHookState OffHook
    Delay 100
    String *95
    ClearCB 2
    JumpTo #00
    SetEvent 2
EndSoftKey
```

```
;-----
;    Label:    Log In
;    Comment: (Idle State) (Log in or out)
;-----
SoftKey=SK_25
    Label=Log In
    SetState 3
    SetInfoFlags 11
    JumpTo LogIn
    SetEvent 1
EndSoftKey
```

```
;-----
;    Label:    OK
;    Comment: (Idle State) Log in or out OK
;-----
SoftKey=SK_26
    Label=OK
    SetInfoFlags 11
    JumpTo LogIn2
    SetEvent 1
EndSoftKey
```

```

;-----
;   Label:   OK
;   Comment: (Idle State) Log in or out OK OK
;-----
SoftKey=SK_27
  Label=OK
  SetInfoFlags 3
  SetHookState OffHook
  Delay 100
  String *98
  Keycode SendChars
  Keycode ClearChars
  ClearCB 2
  JumpTo HangupScreenCE1andInConfCE2
  SetEvent 1
EndSoftKey

```

```

;-----
;   Label:   Mail
;   Comment: (Idle State)
;-----
SoftKey=SK_28
  Label=Mail
  SetState 3
  SetInfoFlags 11
  JumpTo Mail1
  SetEvent 1
EndSoftKey

```

```

;-----
;   Label:   OK
;   Comment: (Idle State) Mail OK
;-----
SoftKey=SK_29
  Label=OK
  SetInfoFlags 11
  JumpTo Mail2
  SetEvent 1
EndSoftKey

```

```

;-----
;   Label:   OK
;   Comment: (Idle State) Mail OK OK
;-----

```

```
SoftKey=SK_30
  Label=OK
  SetInfoFlags 3
  SetHookState OffHook
  Delay 100
  String *99
  Keycode SendChars
  Keycode ClearChars
  String #
  ClearCB 2
  JumpTo HangupScreenCE1andInConfCE2
  SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   Status
;   Comment: (Idle State)
;-----
SoftKey=SK_31
  Label=Status
  SetState 3
  SetInfoFlags 11
  JumpTo Status1
  SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   OK
;   Comment: (Idle State) Status OK
;-----
SoftKey=SK_32
  Label=OK
  SetInfoFlags 11
  JumpTo Status2
  SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   OK
;   Comment: (Idle State) Status OK OK
;-----
SoftKey=SK_33
  Label=OK
  SetInfoFlags 3
  SetHookState OffHook
  Delay 100
```

```
String *99
Keycode SendChars
Keycode ClearChars
String #
String 5
String 1
ClearCB 2
JumpTo HangupScreenCE1andInConfCE2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: CallFwd
; Comment: (Idle State)
;-----
SoftKey=SK_34
Label=CallFwd
SetState 3
SetInfoFlags 11
JumpTo CallFwd1
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) Call Forward OK
;-----
SoftKey=SK_35
Label=OK
SetInfoFlags 11
JumpTo CallFwd2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) Call Forward OK OK
;-----
SoftKey=SK_36
Label=OK
SetInfoFlags 3
SetHookState OffHook
Delay 100
String *99
Keycode SendChars
Keycode ClearChars
```

```
String #
String 5
String 2
ClearCB 2
JumpTo HangupScreenforCallFwd
SetEvent 1
EndSoftKey
```

```
;-----
; Label: DND
; Comment: (Idle State)
;-----
SoftKey=SK_37
Label=DND
SetState 3
SetInfoFlags 11
JumpTo DND1
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) DND OK
;-----
SoftKey=SK_38
Label=OK
SetInfoFlags 11
JumpTo DND2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) DND OK OK
;-----
SoftKey=SK_39
Label=OK
SetInfoFlags 3
SetHookState OffHook
Delay 100
String *99
Keycode SendChars
Keycode ClearChars
String #
String 5
String 1
```

```
String 5
ClearCB 2
JumpTo HangupScreenCE1andInConfCE2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: VMail
; Comment: (Idle State)
;-----
SoftKey=SK_40
Label=VMail
SetState 3
SetInfoFlags 11
JumpTo VMail1
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) VMail OK
;-----
SoftKey=SK_41
Label=OK
SetInfoFlags 11
JumpTo VMail2
SetEvent 1
EndSoftKey
```

```
;-----
; Label: OK
; Comment: (Idle State) Vmail OK OK
;-----
SoftKey=SK_42
Label=OK
SetInfoFlags 3
SetHookState OffHook
Delay 100
String *99
Keycode SendChars
Keycode ClearChars
String #1
ClearCB 2
JumpTo HangupScreenCE1andInConfCE2
SetEvent 1
EndSoftKey
```

```
;-----
;   Label:   More
;   Comment: (Idle State) More to Idle State 2
;-----
```

```
SoftKey=SK_43
    Label=More
    JumpTo IdleState2
    SetEvent 1
EndSoftKey
```

```
;-----
; Primary Script (referred to as Triggers in the menu tree doc)
;-----
```

```
OnEvent Answer
    SetEvent 2
EndEvent
```

```
OnEvent NormalRinging
    ClearDisplay
    SetDisplay NULL,0,0
    SetDisplay Line12,2,3
    SetSoftKey 0,SK_01,Normal
    SetState 1
EndEvent
```

```
; (This section is for the code for the Slot 2 exit upon ring version. I added this
code because I have no examples
; of how an event could trigger a script to exit allowing programming in the main
telco slot to take over (slot 1).
; In this case there is a 3 second delay that I added so as to allow a text line to
show up telling you this is
; where it exits. This delay may prevent slot 1 from becoming the active slot. This
is theoretical to me still,
; since I've not personally had the need to do this in a script before.)
```

```
;
; OnEvent NormalRinging
;     ClearDisplay
;     SetDisplay NULL,0,0
;     SetDisplay Line45,2,3
;     SetTimer Start,3
;     Exit Display           ; Exits and Displays Idle State Screen
;     Exit DoNotDisplay ; Exits but doesn't Display Idle State Screen
```

```

OnEvent CustomEvent1,1
    ClearDisplay
    SetDisplay NULL,0,0
    SetDisplay Line01,1,1
    SetDisplay Line02,2,1
    SetDisplay Line03,3,1
    SetDisplay Line04,4,1
    SetDisplay Line05,5,1
    SetDisplay Line06,6,1
    SetDisplay Line07,7,1
    SetDisplay Line08,8,3
    SetSoftKey
0,SK_24,Normal,SK_40,Normal,SK_37,Normal,SK_34,Normal,SK_31,Normal,SK_43,Normal
    SetState 4
EndEvent

OnEvent CustomEvent2,1|4
    ClearDisplay
    SetDisplay NULL,0,0
    SetDisplay Line13,2,1
    SetDisplay Line15,3,3
    SetSoftKey
0,SK_13,Normal,SK_16,Normal,SK_19,Normal,SK_02,Normal,SK_03,Normal,SK_23,Normal
EndEvent

OnEvent Pickup,1
    SetEvent 1
EndEvent

OnEvent SelectService
    SetState 1
    SetEvent 1
EndEvent

OnEvent CallerID
    ClearDisplay
    SetDisplay NULL,0,0
    SetDisplay Line12,2,1
    SetDisplay Line15,3,3
    SetSoftKey 0,SK_01,Normal
EndEvent

OnEvent EndofRinging
    SetEvent 1
EndEvent

OnEvent FarEndRinging
    SetEvent 2
EndEvent

```

```

    OnEvent FarEndAnswer
        SetEvent 2
    EndEvent

    OnEvent FarEndBusy
        SetEvent 2
    EndEvent

;-----
; **COMMENTS**
;
; The subscripts below are not triggered by events but are ones that we created
; to store things like commands to display softkeys and display text. This way
; we can call softkeys and text and conserve the # of bytes used in a softkey
; by calling a subscript that contains them. Softkeys have 20 bites each. Most
; commands are worth 1 to 2 bites. Each softkey is worth a bite each as is each
; line of text. By putting these into subscripts it only takes 3 bites to call.
;     For example:
;         "JumpTo Mail1"
;         "SetEvent 1"
; It jumps to the subscript named Mail1 and tells it to Set Event 1. If you look
; in the Mail1 subscript below, you see it looks for Event 1.
;     For example:
;         "OnEvent CustomEvent1"
;         "    ClearDisplay"...
; Subscripts are also great for keeping things organized.
;-----

;-----
;     IdleState2
;-----
Subscript IdleState2

    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line02,2,1
        SetDisplay Line11,3,3
        SetSoftKey 0,SK_28,Normal,<NULL>,<NULL>,SK_25,Normal,<NULL>,SK_05,Normal
    EndEvent

EndSubscript

;-----
;     ConnectedState2
;-----
Subscript ConnectedState2

```

```

    OnEvent CustomEvent1
        SetState 5
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line09,2,1
        SetDisplay Line11,3,1
        SetDisplay Line09,4,3
        SetSoftKey
0,SK_22,Normal,SK_21,Normal,<NULL>,SK_02,Normal,SK_03,Normal,SK_08,Normal
        SetTimer Start,3
    EndEvent

    OnEvent Timer,5
        SetState 6
        SetDisplay Line13,2,1
        SetDisplay Line15,3,3
        SetTimer Start,3
    EndEvent

    OnEvent Timer,6
        SetState 5
        SetDisplay Line09,2,1
        SetDisplay Line11,3,1
        SetDisplay Line09,4,3
        SetTimer Start,3
    EndEvent

EndSubscript

;-----
;   Mail1
;-----
Subscript Mail1

    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line26,2,3
        SetSoftKey 0,SK_29,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_06,Normal
    EndEvent

EndSubscript

;-----
;   Mail2
;-----
Subscript Mail2

```

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line27,2,3
  SetSoftKey 0,SK_30,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_06,Normal
EndEvent
```

EndSubscript

```
;-----
;   VMail1
;-----
```

Subscript VMail1

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line26,2,3
  SetSoftKey 0,SK_41,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
EndEvent
```

EndSubscript

```
;-----
;   VMail2
;-----
```

Subscript VMail2

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line27,2,3
  SetSoftKey 0,SK_42,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
EndEvent
```

EndSubscript

```
;-----
;   Status2
;-----
```

Subscript Status2

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line27,2,3
  SetSoftKey 0,SK_33,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
EndEvent
```

EndSubscript

```
;-----  
;   Status1  
;-----
```

Subscript Status1

```
    OnEvent CustomEvent1  
        ClearDisplay  
        SetDisplay NULL,0,0  
        SetDisplay Line26,2,3  
        SetSoftKey 0,SK_32,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal  
    EndEvent
```

EndSubscript

```
;-----  
;   LogIn  
;-----
```

Subscript LogIn

```
    OnEvent CustomEvent1  
        ClearDisplay  
        SetDisplay NULL,0,0  
        SetDisplay Line26,2,3  
        SetSoftKey 0,SK_26,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_06,Normal  
    EndEvent
```

EndSubscript

```
;-----  
;   LogIn2  
;-----
```

Subscript LogIn2

```
    OnEvent CustomEvent1  
        ClearDisplay  
        SetDisplay NULL,0,0  
        SetDisplay Line27,2,3  
        SetSoftKey 0,SK_27,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_06,Normal  
    EndEvent
```

EndSubscript

```
;-----  
;   ConfSelect  
;-----
```

Subscript ConfSelect

```
OnEvent CustomEvent1
  SetState 3
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line43,2,1
  SetDisplay Line24,3,3
  SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
EndEvent
```

EndSubscript

```
;-----
;   Conf
;-----
Subscript Conf
```

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line23,2,3
  SetSoftKey 0,SK_17,Normal,<NULL>,<NULL>,SK_18,Normal,<NULL>,SK_08,Normal
EndEvent
```

EndSubscript

```
;-----
;   CallFwd2
;-----
Subscript CallFwd2
```

```
OnEvent CustomEvent1
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line27,2,3
  SetSoftKey 0,SK_36,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
EndEvent
```

EndSubscript

```
;-----
;   HangupScreenforCallFwd
;-----
Subscript HangupScreenforCallFwd
```

```
OnEvent CustomEvent1
  SetState 3
  ClearDisplay
  SetDisplay NULL,0,0
  SetDisplay Line28,2,1
```

```
        SetDisplay Line29,3,3
        SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
    EndEvent
```

EndSubscript

```
;-----
;   CallFwd1
;-----
```

Subscript CallFwd1

```
    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line26,2,3
        SetSoftKey 0,SK_35,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
    EndEvent
```

EndSubscript

```
;-----
;   HangupScreenCE1andInConfCE2
;-----
```

Subscript HangupScreenCE1andInConfCE2

```
    OnEvent CustomEvent1
        SetState 3
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line22,2,3
        SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
    EndEvent
```

```
    OnEvent CustomEvent2
        SetState 3
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line25,2,3
        SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
    EndEvent
```

EndSubscript

```
;-----
;   DND1
;-----
```

Subscript DND1

```
    OnEvent CustomEvent1
```

```
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line26,2,3
        SetSoftKey 0,SK_38,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
    EndEvent
```

EndSubscript

```
;-----
;   DND2
;-----
```

Subscript DND2

```
    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line27,2,3
        SetSoftKey 0,SK_39,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_05,Normal
    EndEvent
```

EndSubscript

```
;-----
;   Transfer
;-----
```

Subscript Transfer

```
    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line18,2,3
        SetSoftKey 0,SK_14,Normal,<NULL>,<NULL>,SK_15,Normal,<NULL>,SK_08,Normal
    EndEvent
```

EndSubscript

```
;-----
;   TransferCurrentCall
;-----
```

Subscript TransferCurrentCall

```
    OnEvent CustomEvent1
        SetState 3
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line39,2,1
        SetDisplay Line40,3,3
        SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
    EndEvent
```

```

EndSubscript

;-----
;   TransferSelectCall
;-----
Subscript TransferSelectCall

    OnEvent CustomEvent1
        SetState 3
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line41,2,1
        SetDisplay Line42,3,3
        SetSoftKey 0,<NULL>,<NULL>,<NULL>,<NULL>,SK_03,Normal
    EndEvent

EndSubscript

;-----
;   NewCall
;-----
Subscript NewCall

    OnEvent CustomEvent1
        ClearDisplay
        SetDisplay NULL,0,0
        SetDisplay Line19,2,3
        SetSoftKey 0,SK_20,Normal,<NULL>,<NULL>,SK_10,Normal,<NULL>,SK_08,Normal
    EndEvent

EndSubscript

```

A copy of the menu tree for the “Sayson ADSI type 2” text file is also available from Sayson.

FAQ

The Asterisk is not receiving my ADSI scripts. “ADSI Unavailable on CPE. Not bothering to try.” Error message.

Each channel that a phone is on needs to be enabled for ADSI. To do this, add the line “adsi=yes” to the “zapata.conf” file.

What is the difference between ADSI type 1 and ADSI type 2?

The main difference is that in ADSI type 2 there are approximately 62 unique softkeys and 62 unique display lines of text, where as in ADSI type 1 there are only approximately 32 unique softkeys and 32 unique display lines of text. 390 and 480 phones adhere to ADSI type 2 (which is inclusive of ADSI type 1). The older and discontinued phone, the 350, only adhered to ADSI type 1.

How can I restore 390 & 480 phones to their factory state?

Factory Reset for 390 phones:

Please note that this procedure will also erase all entries in the caller's log, the redial list, and the directory.

1. Press the "Options" button, scroll down to option 2 Time/Date and press the "Show" softkey.
2. Change the Time and Date to Jan 1 12:00am, press the "Done" softkey twice.
3. Press the "Options" button again to get back into the options menu.
4. Press the "Mute" button you should see something that resembles the following:

```
FW 99.99
CPE 999A99A9
XX XX
```

5. Then press the "#" button on the dial pad. If the reset is successful you will see a "Please wait..." message and then the following:

In English

```
- Français          -
- Español           -
-                   - Done -
```

6. Press the "Done" softkey to continue in English.

Factory Reset for 480 phones:

Please note that this procedure will also erase all entries in the caller's log, the redial list, and the directory.

1. Press the "Options" button, scroll down to option 2 Time/Date and press the "Show" softkey.
2. Change the Time and Date to Jan 1 12:00am, press the "Done" softkey twice.
3. Press the "Options" button again to get back into the options menu.
4. Press the "Hold" button and then the number "9" and you should see something that resembles the following:

```
FW 99-99-XXXX
CPE 999A99A9
XX XX
```

5. Then press the "#" button on the dial pad. If the reset is successful you will see a "Please wait..." message and then the following:

In English

```
- Français      -
- Español      -
-             Done -
```

6. Press the "Done" softkey to continue in English.